

A Novel Recommender System Fusing the Opinions from Experts and Ordinary People

Pei Wu
Physics Department
University of Fribourg
1700 Fribourg, Switzerland
pei.wu@unifr.ch

Weiping Liu
Physics Department
University of Fribourg
1700 Fribourg, Switzerland
weiping.liu@unifr.ch

Cihang Jin
Physics Department
University of Fribourg
1700 Fribourg, Switzerland
cihang.jin@gmail.com

ABSTRACT

In this paper, we propose a novel recommendation algorithm fusing the opinions from experts and ordinary people. Instead of regarding one's judgement capability as his/her expertise, we present a new definition which measures the amount of the recommendable items one know in a certain area. When computing the expertise, we consider both the average value and the accumulative value, and introduce a free parameter α to tune between these two values. To evaluate the proposed algorithm, simulations are run on the Moviepilot dataset, and the results demonstrate that our algorithm outperforms the conventional collaborative filtering algorithm.

Categories and Subject Descriptors

H.4.m [Information Systems Applications]: Miscellaneous

General Terms

Algorithms

Keywords

recommender systems, experts finding, collaborative filtering

1. INTRODUCTION

The past decade witnesses the explosive growth of the information in the World Wide Web and the emerging of the information overloading problem. When being provided much more information than they are able to process, people can hardly identify what they are looking for. *Information retrieval* and *information filtering* are both important technologies developed to solve this problem. Besides them, recommender systems are another strong candidate to provide automatical and personalized information service, and draw much attention in recent years.

Currently the research on the recommendation algorithm mainly focuses on two problems: the *score-prediction* and the *preference-based filtering* [1]. The goal of the score-prediction is to predict the absolute values of the ratings for the items which one has not seen yet. On the other hand, the preference-based filtering algorithms try to dig out the items which one may like. Obviously, the preference-based filtering algorithms meet better the requirements from the real world, and we will focus on them in this paper.

Many algorithms have been developed to address the personalized recommendation problem, and a good survey can be found in [2][3]. User-based collaborative filtering (CF) is widely used, and the main idea is to find the items liked by other people with similar taste. Different from the user-based CF, the item-based CF recommends the items which are similar with the user's collected items [4][5].

In real world, users carry heterogeneous expertise in real world [6], which leads to different confidence levels of their ratings. If one is an expert in a certain area, his/her ratings are more trustworthy than the ratings of ordinary people. Therefore, emphasizing the experts' ratings is definitely helpful to improve the recommendation accuracy. An iterative refinement method was developed in [6] to estimate users' judgement capabilities. Though it has been shown that the iterative refinement method outperforms the naive method on a synthetic dataset, we argue that the definition of expertise in [6] is not suitable in recommender systems. The iterative refinement algorithm regards the capability of giving correct ratings as the expertise. However, it is intuitive that an expert in recommendation will not only give correct ratings, but also know large amount of recommendable items in his/her area. Actually, for the preference-based filtering, the capability of knowing recommendable items is more important than the capability of giving correct ratings, because the final task of the preference-based filtering is to recommend good items.

In this paper, we propose a recommendation algorithm to fuse the opinions from experts and ordinary people. A new definition of expertise is presented to measure users' knowledge of recommendable items in a certain area. By being assigned larger weights, the opinions from experts will be emphasized in the recommendation. A free parameter β is introduced to trade off between experts' and ordinary people's opinions. Another improvement to the iterative refinement algorithm in [6] is that we use a parameter α to trade off between the average value and the accumulative value when calculating users' expertise and items' qualities. By tuning α , we find the optimal is achieved when considering both the

average value and the accumulative one. We simulate the proposed algorithm on a movie rating dataset from Moviepi-
lot. The results show that our algorithm outperforms the
conventional CF algorithm. When compared to the itera-
tive refinement algorithm, the proposed algorithm provides
higher Area-Under-the-Curve (*AUC*) but lower Mean-Average-
Precision (*MAP*).

2. PRELIMINARIES

In this paper, we consider an explicit rating data con-
sisting of M users and N items. The symbols u and v are
reserved for the indices of users, and i and j are reserved for
items. The ratings matrix is denoted as \mathbf{R} , and the element
 r_{vi} of \mathbf{R} is the rating assigned by user v to item i . The r_{vi}
is either a real value or \emptyset , which means that user v doesn't
assign any rating to item i . To recommend items to user
 u , the algorithm need to sort all the items with $r_{ui} = \emptyset$ by
their probabilities of being liked by user u . The list will be
truncated and the items at the top are delivered to user u
as the recommendation results.

Hence the recommendation algorithm is split into two
parts. In the first part, the probability of being liked by
user u is calculated for each item, and in the second part all
items are sorted by the calculated probabilities. Actually
the second part is trivial, so we mainly focus on the first
part in this paper. In conventional CF, the predicted rating
 \hat{r}_{ui} , namely the probability of item i being liked by user u ,
is calculated as [1]

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathbf{U}_i} s_{uv} r_{vi}}{\sum_{v \in \mathbf{U}_i} s_{uv}}, \quad (1)$$

where \mathbf{U}_i is the set of users who rate item i . s_{uv} , defined
as the similarity between user u and user v , is the measure
of how similar the two users' interests and rating habits are.
The large value of s_{uv} indicates that they are like mind users.
The predicted rating \hat{r}_{ui} is a weighted average of all the avail-
able ratings on item i , and the weights are the similarities.
Therefore the user with larger similarity to the target user u
will have more influences on the predicted ratings, and then
have more influences on the final recommendation results.
In CF framework, the definition of similarity is an impor-
tant and open part, and there are several popular similarity
metrics. Among them, a widely-used similarity, cosine sim-
ilarity, is adopted in this paper, which is defined as

$$s_{uv} = \frac{\vec{r}_u \cdot \vec{r}_v}{\|\vec{r}_u\| \|\vec{r}_v\|} = \frac{\sum_{i \in \mathbf{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathbf{I}_{uv}} r_{ui}^2} \sqrt{\sum_{i \in \mathbf{I}_{uv}} r_{vi}^2}}, \quad (2)$$

where \mathbf{I}_{uv} is the set of the items to whom both user u and
user v assign ratings [1][3].

Besides the cosine similarity, there are still other similar-
ity metrics existing, and some metrics even provide better
performance than the cosine similarity [7]. But finding the
best similarity metric is not the focus of this paper, and our
algorithm can easily substitute the cosine similarity with
other ones. So we only run the simulation with the cosine
similarity, and this won't bring problems in drawing the con-
clusion.

3. FINDING THE EXPERTS

Equation (1) implies that the conventional CF algorithm
weighs users only by how much they are like to the target

user u . However, the users are also different in their ex-
pertise. But this difference is not taken into account in the
conventional CF algorithm, which misses out on the benefits
from experts' reliable opinions. It's obvious that we should
assign large weights to the experts, and low or zero weights
to the ordinary people.

To distinguish the experts and the ordinary people, we
must define a measure of the their expertise. A judgement-
capability-based(JCB) expertise is defined in [6], which em-
phasizes the people who provide the ratings close to the
average ratings. At the same time, the average ratings are
obtained by all users' ratings and their expertise. The peo-
ple with larger expertise will contribute more to the average
ratings. An iterative refinement algorithm has been devel-
oped to calculate the JCB expertise. The formulas in each
iteration read

$$\begin{aligned} r_i^{t+1} &= \sum_{u=1}^M c_u^t r_{ui} \quad \text{for } i = 1 \dots N \\ V_u &= \sqrt{\frac{1}{M} \sum_{i=1}^N (r_{ui} - r_i^t)^2} \quad \text{for } u = 1 \dots M \\ c_u^{t+1} &= \frac{1/\sqrt{V_u}}{\sum_{v=1}^M 1/\sqrt{V_v}} \quad \text{for } u = 1 \dots M, \end{aligned} \quad (3)$$

where t is the index of the iteration step, r_i^t is the average
ratings of item i in step t , c_u^t is the expertise of user u in
step t , and V_u is a temporary variable.

Though the JCB expertise and iterative refinement method
work well on the synthetic dataset in [6], they are impracti-
cal in the real world. In real world, users won't assign ratings
to all items, and the number of the items to which a certain
user assigns ratings, namely the degree of that user, shows
larger diversity. And so does the degree of the items. These
lead to unexpected results when applying equation (3). For
example, if a certain user u only rates item i , and item i
is only rated by user u . In such case, the average rating of
item i is exactly r_{ui} , so V_u equals to zero, which makes equa-
tion (3) fall into meaninglessness. Another problem of the
JCB expertise is that it only takes into account the average
value when calculating the expertise. It's inappropriate in
real systems. Considering two users with different degree,
obviously the one with larger degree should be paid more
attentions to even if he/she makes the same average errors
in rating as the other user does.

Besides the discussion above, the JCB expertise regards
one's capability of giving correct ratings as his/her expertise.
We argue that it is not competent in recommender systems,
because the experts are not the rating machines but the
people who can provide many recommendable items. Based
on this consideration, we define e_u , the expertise of user u ,
as

$$e_u = \frac{\sum_{i \in \mathbf{I}_u} q_i}{(D_u)^\alpha}, \quad (4)$$

where \mathbf{I}_u is the set of the items rated by the user u , q_i is
the quality of item i , and D_u is the degree of user u . α is
a free parameter. When $\alpha = 1$, equation (4) only takes into
account the average value. In other words, e_u is the average
quality of the items rated by user u . When $\alpha = 0$, only
the accumulative value is considered and e_u is the sum of
the quality of all items rated by user u . In such case, the
user rating more items will be assigned larger expertise. By

tuning α between 0 and 1, we can make tradeoff between considering the average value and accumulative value.

Similarly, the quality of item i is calculated as

$$q_i = \frac{\sum_{u \in U_i} e_u}{(D_i)^\alpha}, \quad (5)$$

where the D_i is the degree of item i . By updating the e_u and q_i iteratively using (4) and (5), we can get the estimated expertise of users when the iteration reaches convergence.

4. FUSING EXPERTS' AND ORDINARY PEOPLE'S OPINIONS TO MAKE RECOMMENDATION

After computing users' expertise, we can put larger weights to the experts' opinions to emphasize them in recommendation. Nevertheless, the ordinary people's opinions are not totally useless. Actually, the ensemble of many ordinary people's opinions is able to give quite accurate prediction. The tradeoff between experts' and ordinary people's opinions is worthy to be investigated to improve the recommendation accuracy.

In this section, we propose an expertise-weighted CF (EWCF) algorithm. By assigning different weights to the users with different expertise, the algorithm fuses the opinions from experts and ordinary people. Denoting the e_{min} as the minimum value of the non-zero e_u , where $u = 1 \dots M$, we define the weights as

$$w_u = \begin{cases} (e_{min}/2)^\beta, & \text{if } e_u = 0 \\ (e_u)^\beta, & \text{otherwise.} \end{cases} \quad (6)$$

And the prediction of \hat{r}_{ui} in equation (1) is modified to

$$\hat{r}_{ui} = \frac{\sum_{v \in U_i} w_v s_{uv} r_{vi}}{\sum_{v \in U_i} w_v s_{uv}}. \quad (7)$$

5. EXPERIMENTAL RESULTS

To evaluate the performance of the EWCF algorithm, we run simulations on the Moviepilot dataset. The dataset consists of 105131 users, 25057 movies and 4544387 ratings. The ratings are integers from 0 to 100. Part of the movies are tagged with a *mood_id* from 1 to 16. Our goal is to recommend the movies tagged with *mood_id* = 16 (as asked by the Challenge on Context-aware Movie Recommendation [8]). To evaluate the algorithm, a test set with 160 users and 2656 ratings are left out and unknown to the recommendation algorithms.

Four performance metrics, $P@5$, $P@10$, MAP and AUC , are computed for evaluation. Equations (8) to (10) give the definitions of the four metrics of an individual user [9][10]. The metrics of the whole system are the average of the individual ones.

$$P@l_u = \frac{\sum_{r=1}^l \text{rel}(l)}{l} \quad (8)$$

$$MAP_u = \frac{\sum_{l=1}^{N_t} P@l_u \times \text{rel}(l)}{N_t} \quad (9)$$

$$AUC_u = \frac{\sum_{l=1}^{N-D_u} (N - D_u - l) \text{rel}(l)}{(N - D_u) \sum_{l=1}^{N-D_u} \text{rel}(l)} \quad (10)$$

In the above equations, $\text{rel}(l)$ is a binary function which equals 1 if the item with rank l in the sorted list is rated by

Table 1: Algorithmic Performance

Algorithms	$P@5$	$P@10$	AUC	MAP
conventional CF	0	0	0.6519	$3.208E-3$
EWCF	0	0	0.6625	$3.488E-3$
IR	0	0	0.6548	$3.712E-3$

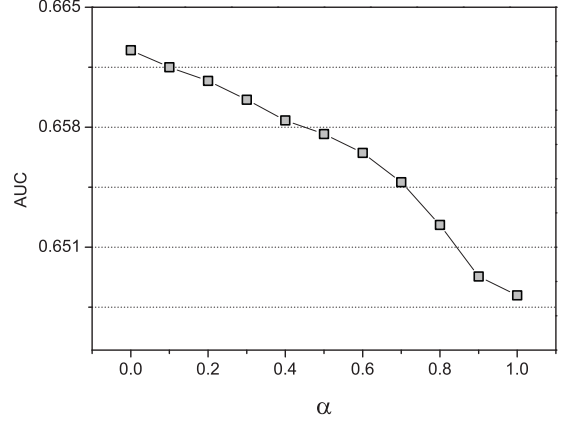


Figure 1: AUC versus α ($\beta = 2.4$)

user u in the test set, and equals 0 otherwise. N_t is the total number of the items in the test set.

To investigate the influence of the parameters α and β on the algorithmic performance, we run simulations for all combinations of $\alpha \in \{0, 0.1, 0.2 \dots 1\}$ and $\beta \in \{0, 0.2, 0.4 \dots 4\}$. Figure 1 and 2 demonstrate the influence of parameter α on AUC and MAP when β is fixed to 2.4. The AUC decreases monotonically with the increasing of α , and the optimal AUC is achieved when $\alpha = 0$, namely only considering the accumulative value. But the situation is different with MAP . The optimal MAP is achieved when $\alpha = 0.2$, which means that considering both average and accumulative values will improve the MAP performance.

The relationship between the performance and the parameter β is shown in Figure 3 and 4. The case of $\beta = 0$ corresponds to the conventional CF, which doesn't distinguish experts and ordinary people. The larger β , the more attentions are paid to the experts' opinions. From Figure 3 and 4, we find that too much weighing or too less weighing the experts will both degrade the performance. To obtain the optimal recommendation accuracy, we must appropriately fuse the opinions from experts and ordinary people.

Finally, we report the optimal AUC and MAP of our algorithm in Table 1. For comparison, we also run the simulations for the iterative refinement (IR) method in [6] and the conventional CF algorithm. It is found that the AUC performance of our algorithm outnumbers the IR method, but the MAP performance of our algorithm is worse. All $P@5$ and $P@10$ of these algorithms are zero.

6. CONCLUSIONS

Computing the different expertise carried by different users gives us the possibility of improving the recommendation accuracy by distinguishing them. The available literature has provide the expertise definition based on users' judgement

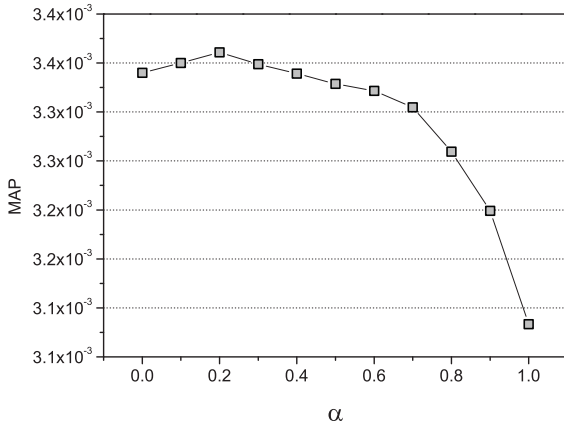


Figure 2: MAP versus α ($\beta = 2.4$)

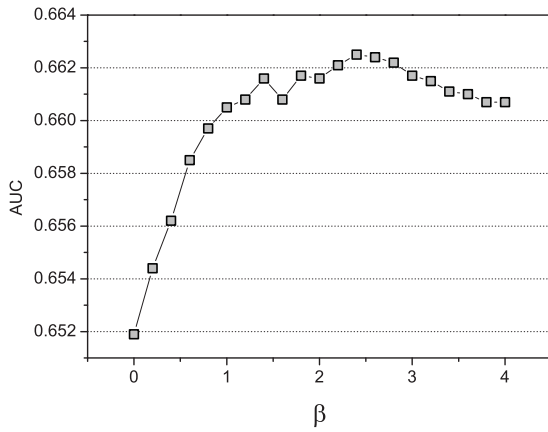


Figure 3: AUC versus β ($\alpha = 0$)

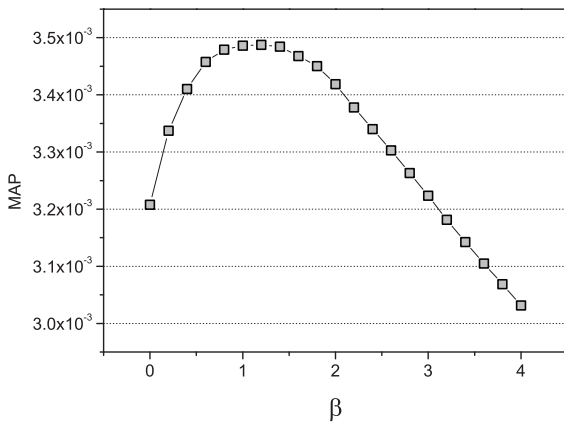


Figure 4: MAP versus β ($\alpha = 0$)

capabilities. In this paper, we argue that users' judgement capabilities are not the key point to the recommendation, and we propose a new definition considering how many recommendable items the users know. Another contribution of this paper is that we trade off between the average value and accumulative value when computing expertise. The evaluation of our algorithm is done on Moviepilot dataset. $P@5$, $P@10$, MAP and AUC are calculated as the evaluation metrics. The results demonstrate that our algorithm outperforms the conventional CF algorithm, and is superior to the IR method with AUC metric.

7. ACKNOWLEDGMENTS

We acknowledge funding from the Liquid Publications (EU FET-Open Grants 213360).

8. REFERENCES

- [1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. Knowl. Data Eng.*, 17(6):734–749, June 2005.
- [2] L. Candillier, K. Jack, F. Fessant, and F. Meyer. State-of-the-art recommender systems. In *Collaborative and Social Information Retrieval and Access - Techniques for Improved User Modeling*, pages 1–22. Idea Group, 2009.
- [3] X. Y. Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009(4), 2009.
- [4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. 10th Int. WWW Conf.*, May 2001.
- [5] M. Deshpande and G. Karypis. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.
- [6] Y. K. Yu, Y. C. Zhang, P. Laureti, and L. Moret. Decoding information from noisy, redundant, and intentionally-distorted sources. *Physica A*, 371(2):732–744, November 2006.
- [7] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 43–52, 1998.
- [8] Alan Said, Shlomo Berkovsky, and Ernesto W. De Luca. Putting things in context: Challenge on context-aware movie recommendation. In *CAMRa2010: Proceedings of the RecSys '10 Challenge on Context-aware Movie Recommendation*, New York, NY, USA, 2010. ACM.
- [9] L. G. Terveen, J. L. Herlocker, J. A. Konstan and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, 2004.
- [10] C. Buckley and E. M. Voorhees. Retrieval system evaluation. In *TREC: experiment and evaluation in information retrieval*. MIT Press, 2005.